Name: Vignesh Iyer
Collaborators:
Bruce Englert
Madison Shaerr
Sarah Marshall
Zach Shepelak
Zhu - Data Project No 1

1. **(Note: 3 from document) Generating a time series for the returns in the following steps:**
   (a) Use a size-based weighted average to list a price for each minute:

$$\frac{\omega_1 x_1+\omega_2 x_2+\omega_3 x_3+...+\omega_n x_n}{\omega_1+\omega_2+\omega_3+...+\omega_n}$$

```
##Python Code:
import pandas as pd
import numpy as np
import scipy as sp
from matplotlib import pyplot as plt

df = pd.read_csv("desktop/aapl_copy.csv", sep='|', header=None, parse_dates=True)
df.rename(columns={0: 'date', 1: 'volume', 2: 'price'}, inplace=True)
column was a proxy for volume
df.set_index('date', inplace=True)
df.index = pd.to_datetime(df.index, dayfirst=True) ##specifing date/time
print(df.head())

def wavg(group, price_name, weight_name):
   d = group[price_name]
   w = group[weight_name]
   try:
       return (d * w).sum() / w.sum()
   except ZeroDivisionError:
       return 'NaN'
```

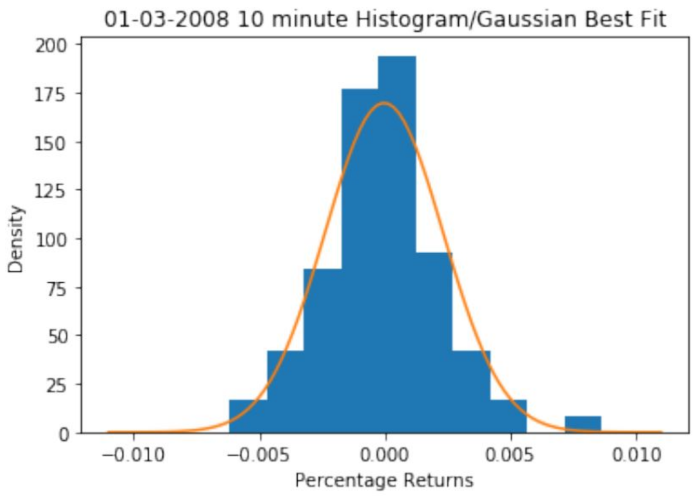
   (b) Use the averaged minutely price series to generate a return series:

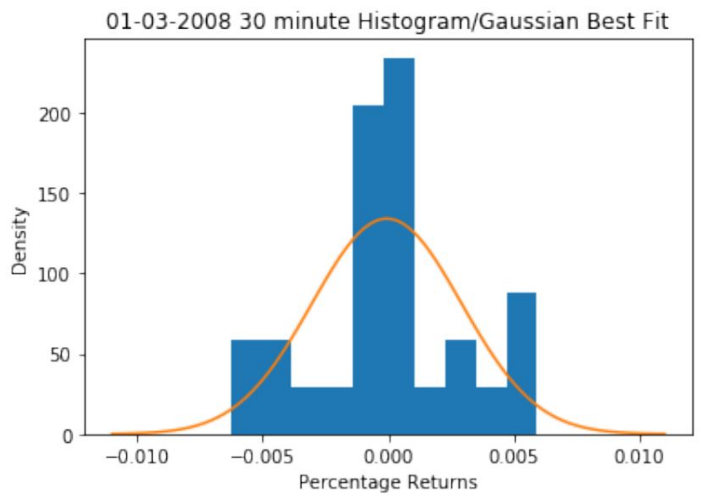
```
##1 minute partition
#wavg_series = df.groupby(pd.Grouper(freq='60s')).apply(wavg, 'price','bulk')
wavg_series = wavg_series.dropna(how='any') ##Continued

print(wavg_series.head())
return_series = wavg_series.pct_change()
return_series = return_series.dropna(how='any')
print(return_series.head())
```

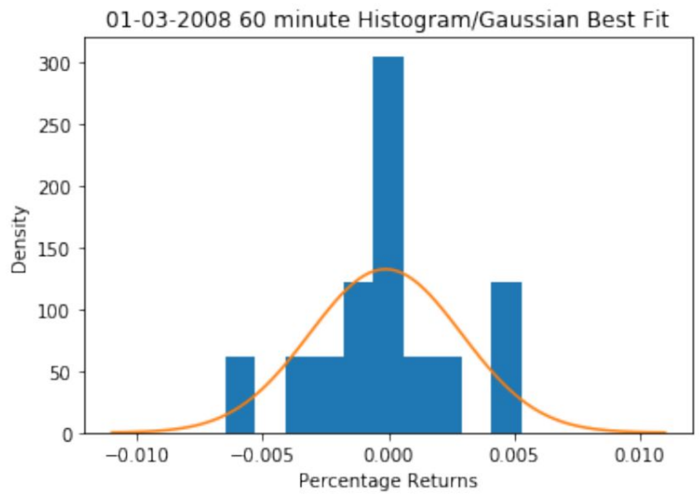
(c) Alternatively, we can use a time period of 10 minutes, 30 minutes, or 60 minutesto generate the returns.

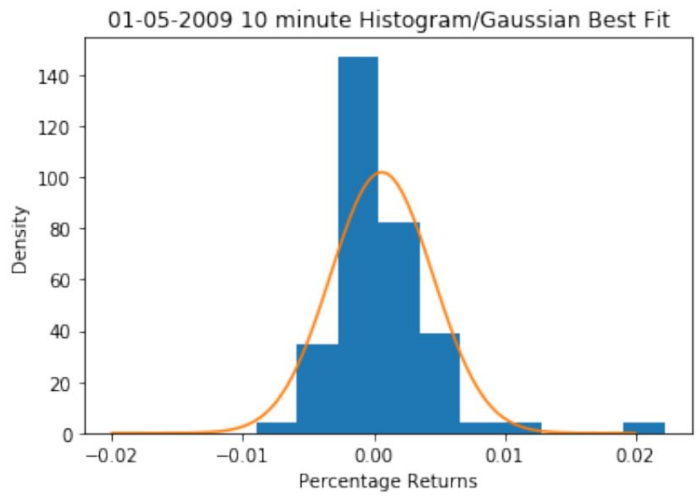
```
#10 minute partition
wavg_series = df.groupby(pd.Grouper(freq='10min')).apply(wavg, 'price','bulk')

#30 minute partition
wavg_series = df.groupby(pd.Grouper(freq='30min')).apply(wavg, 'price','bulk')

#60 minute partition
wavg_series = df.groupby(pd.Grouper(freq='1h')).apply(wavg, 'price','volume')
```

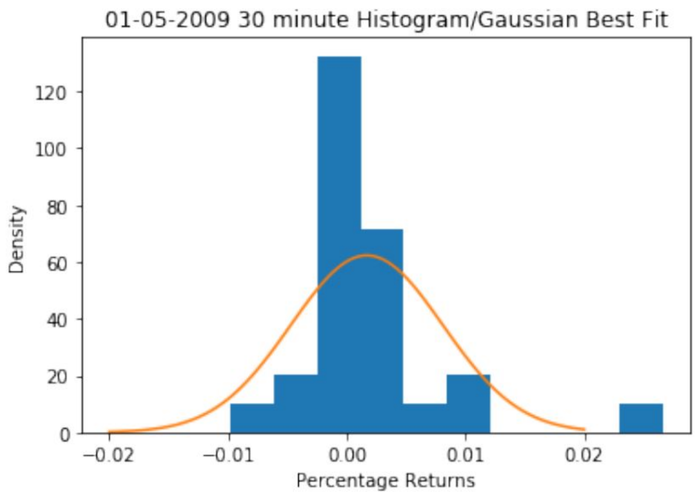
2. **(Note: 4 from document) In this part, we generate histograms for the returns:**
   Analysis on final page.
   (a) Select one day for each year in the dataset, display the histograms for the returns for each of the three time interval cases: *Following histograms shown below:*

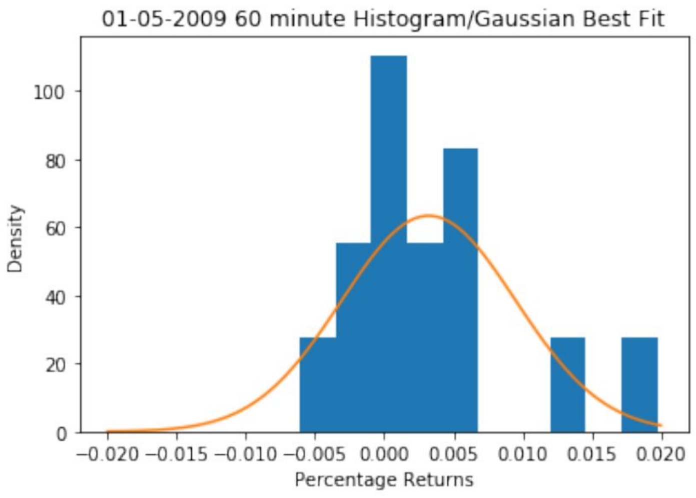


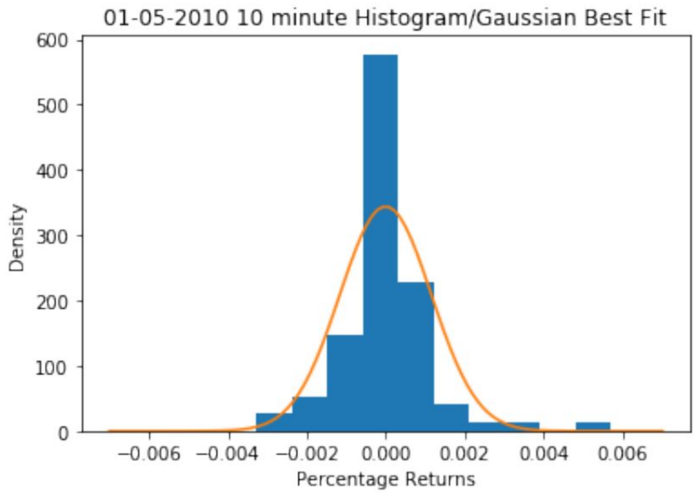

01-03-2008 30 minute Histogram/Gaussian Best Fit
Density
Percentage Returns
0
50
100
150
200
−0.010
−0.005
0.000
0.005
0.010

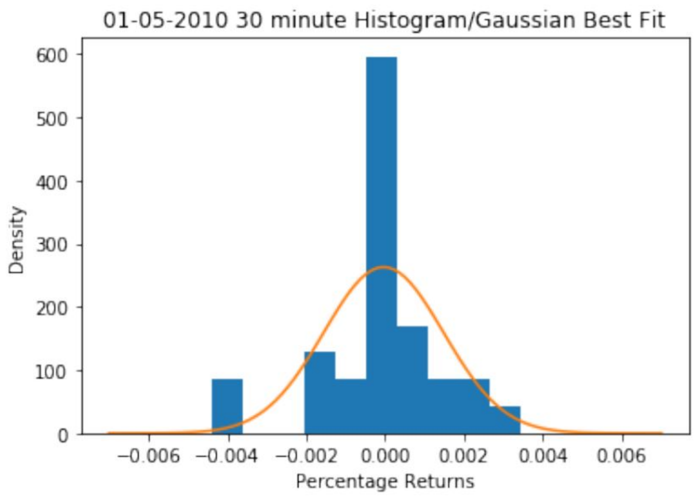

01-03-2008 60 minute Histogram/Gaussian Best Fit
Density
Percentage Returns
0
50
100
150
200
250
300
−0.010
−0.005
0.000
0.005
0.010

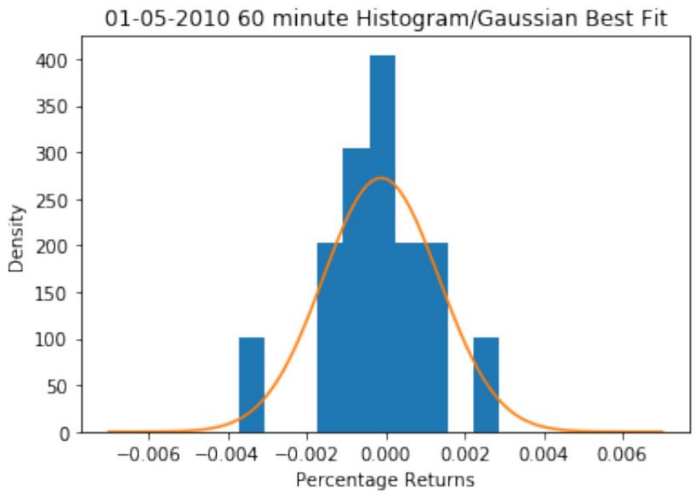

01-05-2009 10 minute Histogram/Gaussian Best Fit
Density
Percentage Returns
0
20
40
60
80
100
120
140
−0.02
−0.01
0.00
0.01
0.02

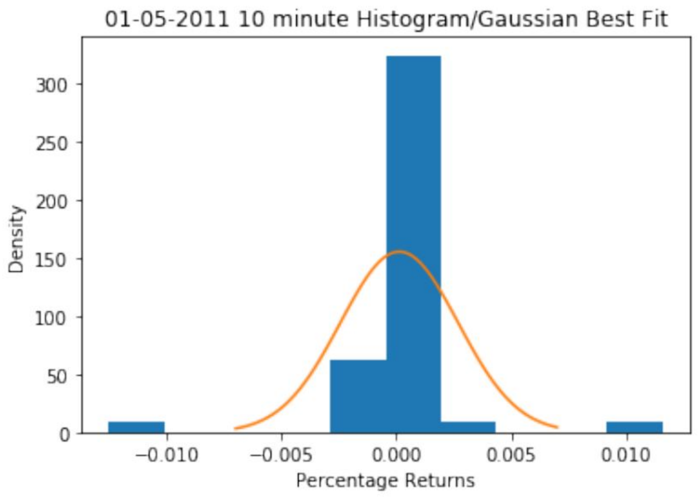

01-05-2009 30 minute Histogram/Gaussian Best Fit
Density
Percentage Returns
0
20
40
60
80
100
120
−0.02
−0.01
0.00
0.01
0.02

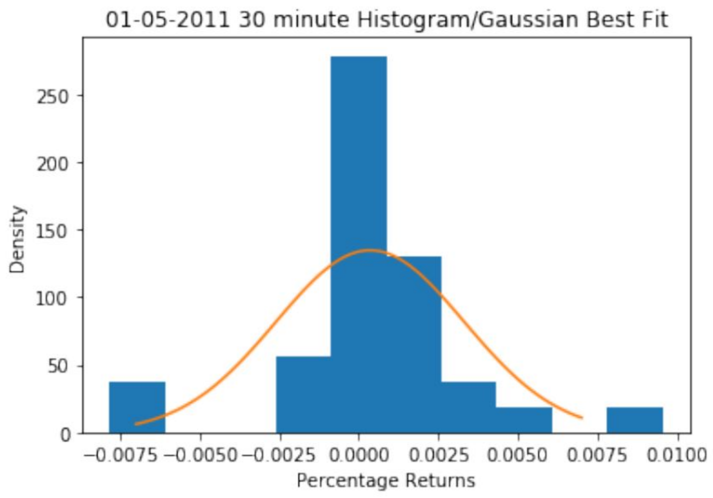

01-05-2009 60 minute Histogram/Gaussian Best Fit
Density
Percentage Returns
0
20
40
60
80
100
−0.020
−0.015
−0.010
−0.005
0.000
0.005
0.010
0.015
0.020

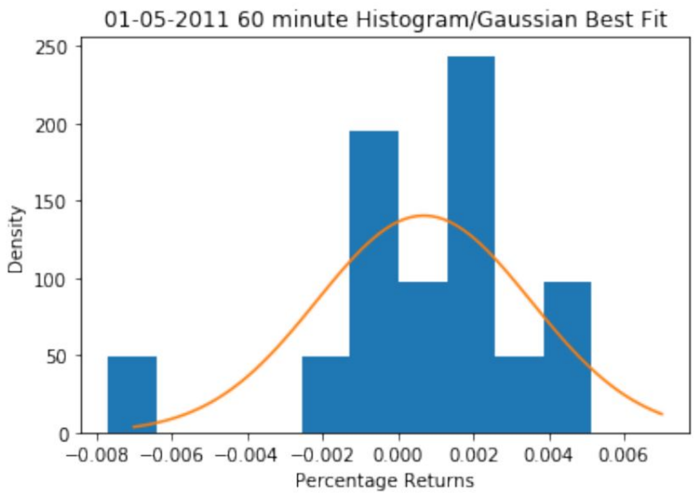

01-05-2010 10 minute Histogram/Gaussian Best Fit
Density
Percentage Returns
0
100
200
300
400
500
600
−0.006
−0.004
−0.002
0.000
0.002
0.004
0.006


01-05-2010 30 minute Histogram/Gaussian Best Fit
Density
Percentage Returns

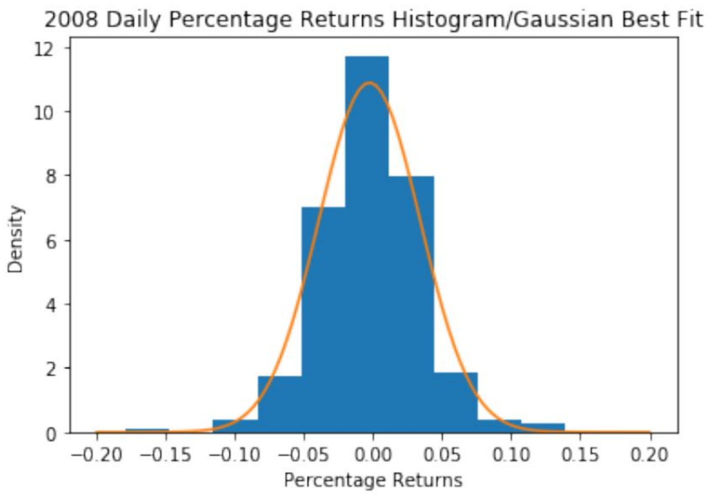

01-05-2010 60 minute Histogram/Gaussian Best Fit
Density
Percentage Returns

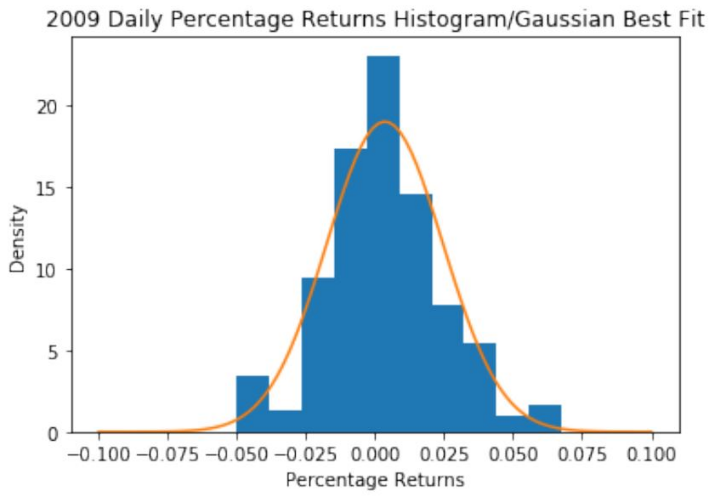

01-05-2011 10 minute Histogram/Gaussian Best Fit
Density
Percentage Returns

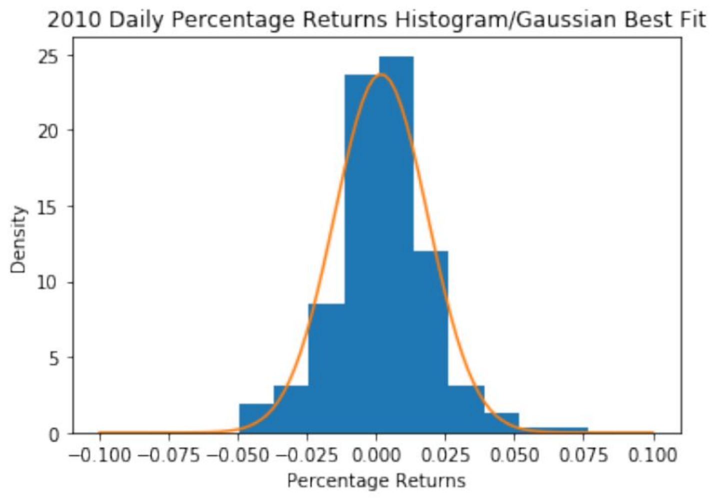

01-05-2011 30 minute Histogram/Gaussian Best Fit
Density
250
200
150
100
50
0
−0.0075
−0.0050
−0.0025
0.0000
0.0025
0.0050
0.0075
0.0100
Percentage Returns

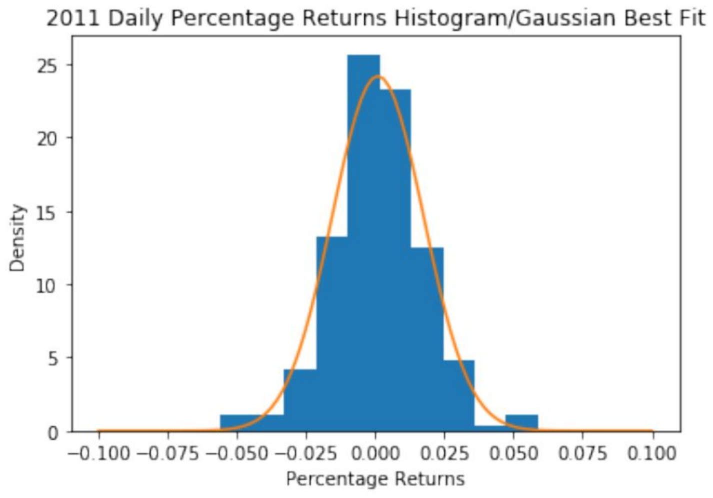

01-05-2011 60 minute Histogram/Gaussian Best Fit
Density
250
200
150
100
50
0
−0.008
−0.006
−0.004
−0.002
0.000
0.002
0.004
0.006
Percentage Returns

(b) Download the daily closing prices for the year in question and compare the histogram of the daily returns with those minutely returns, and comment on the comparisons.






2010 Daily Percentage Returns Histogram/Gaussian Best Fit
Density
Percentage Returns
0
5
10
15
20
25
−0.100
−0.075
−0.050
−0.025
0.000
0.025
0.050
0.075
0.100


2011 Daily Percentage Returns Histogram/Gaussian Best Fit
Density
Percentage Returns
0
5
10
15
20
25
−0.100
−0.075
−0.050
−0.025
0.000
0.025
0.050
0.075
0.100

3. **(Note: 5 from document) Use a MLE function in your program to obtain a best fit Gaussian Distribution for each of the cases in Question 4, and make some comments on the goodness of the fit:**
   Analysis on final page.

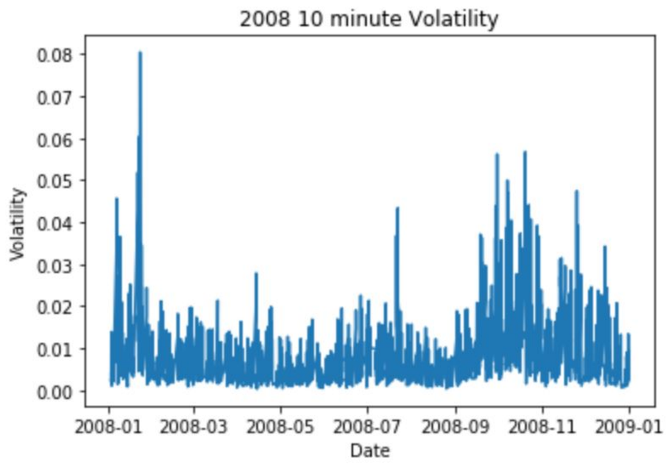
   ```
   from matplotlib import pyplot as plt
   norm.fit(day2008)
   ```

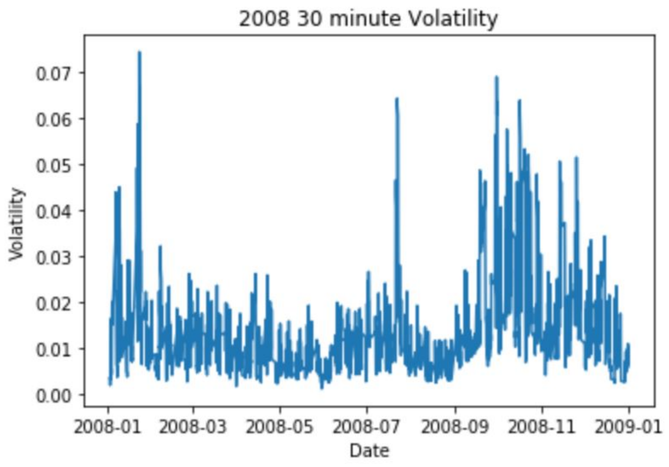
   We chose to use a norm.fit(day2008) in which the parameter passed in (daydate) is a specified return series. The best fit Gaussian Distribution for each time interval for each year is included in the histograms in 2a. See histograms after question 2.

4. **(6 from document) Use the data from 2008 to generate a 10-day exponentially weighted moving average for volatility of the stock. Label those times when the return is higher than 1.5σ or lower than -1.5σ , and experiment with the time intervals of 10-minute and 30-minute for possible trading strategies.**

   ```
   ## Computing Volatility
   def getDailyVol(close,span0 = 10):
       df0 = close.index.searchsorted(close.index-pd.Timedelta(days = 1))
       df0 = df0[df0>0]
       df0 = pd.Series(close.index[df0 - 1], index = close.index
       [close.shape[0]-df0.shape[0]:])
       df0 = close.loc[df0.index]/close.loc[df0.values].values-1 # daily returns
       df0 = df0.ewm(span = span0).std()
       return df0
   df2 = pd.read_csv("desktop/aapl_copy.txt", sep='|', header=None, parse_dates=True)
   df2.rename(columns={0: 'date', 1: 'volume', 2: 'price'}, inplace=True)
   df2.set_index('date', inplace=True)
   df2.index = pd.to_datetime(df2.index, dayfirst=True)
   wavg_series1008 = df2.groupby(pd.Grouper(freq='10min')).
   apply(wavg, 'price','volume')
   wavg_series1008 = wavg_series1008.dropna(how='any')
   wavg_series3008 = df2.groupby(pd.Grouper(freq='30min')).
   apply(wavg, 'price','volume')
   wavg_series3008 = wavg_series3008.dropna(how='any')
   vol1008 = getDailyVol(wavg_series1008)
   vol1008 = vol1008.dropna(how='any')
   vol3008 = getDailyVol(wavg_series3008)
   vol3008 = vol3008.dropna(how='any')
   plt.plot(vol1008)
   plt.title('2008 10 minute Volatility')
   plt.xlabel('Date')
   plt.ylabel('Volatility')
   ```

```
plt.plot(vol3008)
plt.title('2008 30 minute Volatility')
plt.xlabel('Date')
plt.ylabel('Volatility')
```





Analysis: In the smaller time intervals, their exists a higher degree of accuracy for the curve fitting. For our results, the 10 minute time intervals produced the best accuracy. Additionally, larger variance exists as the time interval increases. In terms of the time intervals, the majority of return of the 10, 30, and 60 min intervals were around 0%. For a long term investment, their is some gain but as a short term investment, this is not good. In other words, the stock market is unpredictable.
In comparison to the closing prices, in 2008 the return varied between 0 and 5%. This means that the 10 minute return within that day isn't good information to interpret daily returns over the year.
In 2009, the return varied between 0 and 2.5%. This seems consistent with the 2009 10 min intervals as their were some returns that did have some gain.
In 2010, the return varied between 0 and 2.5%. Over the 10 min interval for the day selected in 2010, the average return was around 0% and less variance in contrast to 2008 and 2009.
In 2011, the return varied between 0 and 2.5%. The specific day chosen in 2011 had higher variance and this is consistent because this was three years after the recession and the market re-stabilized. The 10, 30, and 60 min interval returns were skewed to the right, potentially indicating a positive return for the day. In addition to the daily closing prices for the four years, 2008 had the most variance possibly due to the market crash. As the years progressed, the variance steadily decreased. Average daily return was increasing and variance decreased.
The curve fitting was much better for 10 min intervals as opposed to the 30 and 60 min time intervals for the four years. For the downloaded closing prices, the curve fitting fits accurately with the daily returns.
The volatility graphs were generated for the year 2008 within the 10 and 30 min time interval. What we concluded from these graphs was that the volatility was greater in the 30 min time interval but overall the spike was shown in both.